AMENDMENTS TO THE CLAIMS:

1. (Currently amended) A software method of improving at least one of efficiency and speed in executing a linear algebra subroutine on a computer having a floating point unit (FPU) and a load/store unit (LSU) capable of overlapping loading data and processing of said FPU data by the FPU, said method comprising:

for an execution code controlling operation of ~~a floating point unit (said FPU) performing~~ said linear algebra subroutine execution, ~~unrolling an instruction to preload~~ overlapping by preloading data into ~~a~~ floating point registers (FRegs) of said FPU, said ~~unrolling~~ overlapping causing ~~said instruction to load~~ data to arrive into said FRegs to be ~~inserted into a sequence of instructions that execute~~ timely executed by the FPU operations of said linear algebra subroutine on said FPU.

2. (Currently amended) The method of claim 1, wherein ~~said~~ instructions are unrolled repeatedly until the data loading reaches a steady state in which a data loading exceeds a data consumption.

3. (Original) The method of claim 1, wherein said linear algebra subroutine comprises a matrix multiplication operation.

4. (Original) The method of claim 1, wherein said linear algebra subroutine comprises a subroutine from a LAPACK (Linear Algebra PACKage).

5. (Original) The method of claim 4, wherein said LAPACK subroutine comprises a BLAS Level 3 L1 cache kernel.

6. (Currently amended) An apparatus, comprising:

a memory to store matrix data to be used for processing in a linear algebra program;

an L1 cache to receive data from said memory;

a floating point unit (FPU) to perform said processing; and

a load/store unit (LSU) to load data to be processed by said FPU, said LSU loading said data into a plurality of floating point registers (FRegs),

wherein said data processing overlaps said data loading such that matrix data is preloaded into said FRegs from said L1 cache prior to being required by said FPU.

7. (Original) The apparatus of claim 6, wherein said preloading is achieved by unrolling a loading instruction so that a load occurs every cycle until a preload condition has been satisfied.

8. (Original) The apparatus of claim 6, wherein said linear algebra program comprises a matrix multiplication operation.

9. (Original) The apparatus of claim 6, wherein said linear algebra program comprises a subroutine from a LAPACK (Linear Algebra PACKage).

10. (Original) The apparatus of claim 9, wherein said LAPACK subroutine comprises a BLAS Level 3 L1 cache kernel.

11. (Original) The apparatus of claim 6, further comprising:

a compiler to generate an instruction for said preloading.

12. (Currently amended) A ~~signal bearing~~ machine-readable medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of improving at least one of speed and efficiency in executing a linear algebra subroutine on a computer having a floating point unit (FPU) and a load/store unit (LSU) capable of overlapping loading data and processing said data, said method comprising:

for an execution code controlling operation of ~~a floating point unit (FPU) performing~~ said linear algebra subroutine execution, ~~unrolling an instruction to preload~~ overlapping by preloading data into ~~a~~ floating point registers (FRegs) of said FPU, said ~~unrolling~~ overlapping causing ~~said instruction to load~~ data from an L1 cache to arrive into said FRegs, to be ~~inserted into a sequence of instructions that execute~~ timely executed by FPU operations of said linear algebra subroutine on said FPU.

13. (Currently amended) The ~~signal bearing~~ machine-readable medium of claim 12, wherein ~~said~~ a load instruction is unrolled repeatedly until the data loading reaches a steady state in which a data loading exceeds a data consumption.

14. (Currently amended) The ~~signal bearing~~ machine-readable medium of claim 12, wherein said linear algebra program comprises a matrix multiplication operation.

15. (Currently amended) The ~~signal bearing~~ machine-readable medium of claim 12, wherein said linear algebra program comprises a subroutine from a LAPACK (Linear Algebra PACKage).

16. (Currently amended) The ~~signal-bearing~~ machine-readable medium of claim 15, wherein said LAPACK subroutine comprises a BLAS Level 3 L1 cache kernel.

17. (Currently amended) A method of providing a service involving at least one of solving and applying a scientific/engineering problem, said method comprising at least one of:

using a linear algebra software package that computes one or more matrix subroutines, wherein said linear algebra software package generates an execution code controlling a load/store unit loading data into ~~a~~ floating point registers (FRegs) for a floating point unit (FPU) performing a linear algebra subroutine execution, said FPU capable of overlapping loading data and performing said linear algebra subroutine processing, such that, for an execution code controlling operation of said FPU, ~~an instruction is unrolled~~ said overlapping ~~to cause~~ causes a preloading of data from an L1 cache into said FRegs;

providing a consultation for purpose of solving a scientific/engineering problem using said linear algebra software package;

transmitting a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result; and

receiving a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result.

18. (Original) The method of claim 17, wherein said linear algebra subroutine comprises a subroutine from a LAPACK (Linear Algebra PACKage).

19. (Original) The method of claim 18, wherein said LAPACK subroutine comprises a BLAS Level 3 L1 cache kernel.

20. (New) The method of claim 1, wherein said FPU comprises said FRegs as interfaced with an L1 cache, said interface having a penalty of n cycles, said preloading eliminating this n-cycle penalty.